

Randomized Algorithms: Distance Approximation & Graph Sparsification

Keerti Choudhary

IIT Delhi

CS Theory Winter School, 2023

Outline

Introduction

Distance Approximation

Graph Sparsification

Exercises

Exact Distance Computation

► **Given:** $G = (V, E)$, $n = |V|$, $m = |E|$

► **Find:** Distance matrix M such that

$$\forall i, j \quad M[i, j] = \text{dist}_G(i, j)$$

► **Time:** $O(mn)$ or $O(n^3)$

Distance (+2) Approximation

► **Given:** $G = (V, E)$, $n = |V|$, $m = |E|$

► **Find:** Matrix M such that $\forall i, j$

$$\text{dist}_G(i, j) \leq M[i, j] \leq \text{dist}_G(i, j) + 2$$

► **Goal:** $O(n^{2.5})$ time

Graph Sparsification

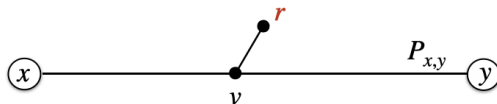
- ▶ **Given:** $G = (V, E)$, $n = |V|$, $m = |E|$
- ▶ **Question:** Sparsify G (remove edges) such that $\forall i, j$
 $dist(i, j)$ is increased by at most $+2$
- ▶ **Goal:** $O(n^{1.5})$ edges

Distance Approximation

Why +2 error

Lemma:

Suppose $x, y \in V$ and $P_{x,y} = (x, y)$ -shortest-path. Also $v \in P_{x,y}$ and $r \in N_G(v)$.



Then,

$$\text{dist}_G(x, r) + \text{dist}_G(r, y) \leq \text{dist}_G(x, y) + 2.$$

Remark: We use this lemma only when $P_{x,y}$ has high degree vertices.

Case 1: All vertices in $P_{x,y}$ have low degree

Notation

$$H = \left(V, \bigcup_{\substack{z \in V \text{ s.t.} \\ \deg(z) < \sqrt{n}}} \text{Edges}(z) \right)$$

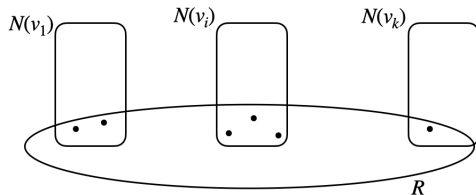
Lemma:

Suppose $x, y \in V$ and $P_{x,y}$ has all vertices of degree at most $\sqrt{n}$.
Then,

$$\text{dist}_G(x, y) = \text{dist}_H(x, y).$$

Hitting Set

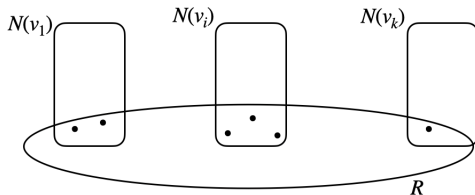
- Let $v_1, \dots, v_k$ be vertices of high degree ($\geq L$).
- Let R be a **random** set obtained by including each vertex independently with probability $p = \frac{1}{L}$.



- Then $E[|N(v_i) \cap R|] \geq 1$, for all i .

Hitting Set

- Let $v_1, \dots, v_k$ be vertices of high degree ($\geq L$).
- Let R be a **random** set obtained by including each vertex independently with probability $p = \frac{4 \log n}{L}$.



- Then $|N(v_i) \cap R| \geq 1$, for all i , with probability 0.9999....

Hitting Set

Proof sketch

For any v_i , probability $|N(v_i) \cap R| = 0$ is

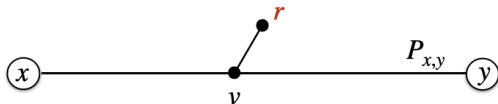
$$\left(1 - \frac{4 \log n}{L}\right)^{|N(v_i)|} \leq \left(1 - \frac{4 \log n}{L}\right)^L \leq e^{-4 \ln n} = \frac{1}{n^4}.$$

Therefore, by Union bound, the probability that $|N(v_i) \cap R| = 0$, for some $i \in [1, k]$, is at most $\frac{1}{n^3}$.

Case 2: At least one vertex in $P_{x,y}$ has high degree

Lemma:

Suppose $P_{x,y}$ has a vertex, say v , of degree more than $\sqrt{n}$, and let R be random set of $O(\sqrt{n} \log n)$ size.



Then,

$$\min_{r \in R} (dist_G(x, r) + dist_G(r, y)) \leq dist_G(x, y) + 2.$$

Algorithm

1. Compute all-pairs distances in $H = \left(V, \bigcup_{\substack{z \in V \text{ s.t.} \\ \deg(z) < \sqrt{n}}} \text{Edges}(z) \right)$
2. Let R be **random** set of size $O(\sqrt{n} \log n)$.
3. For $r \in R$, find $BFS(r)$.
4. For $x, y \in V$:

$$M[x, y] = \min \left(\text{dist}_H(x, y), \min_{r \in R} (\text{dist}_G(x, r) + \text{dist}_G(r, y)) \right)$$

Theorem [DHZ96]

For any n vertex undirected graph G , a $+2$ approximate distance matrix can be computed in $O(n^{2.5} \log n)$ time.

Graph Sparsification

Graph Sparsification

- **Problem:** Find a subgraph H of G such that $\forall i, j$

$$\text{dist}_H(i, j) \leq \text{dist}_G(i, j) + 2$$

- **Goal:** $O(n^{1.5})$ edges

Algorithm

1. Initialise graph $H = \left(V, \bigcup_{\substack{z \in V \text{ s.t.} \\ \deg(z) < \sqrt{n}}} \text{Edges}(z) \right)$
2. Let R be **random** set of size $O(\sqrt{n} \log n)$.
3. For $r \in R$, add edges of $BFS(r)$ to H .
4. Return H .

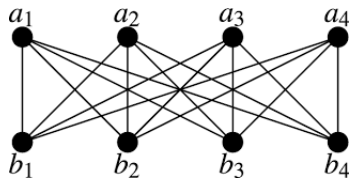
Theorem [DHZ96]

For any n vertex G , we can find an $O(n^{1.5} \log n)$ sized **distance-sparsifier** with additive error $+2$ in $O(n^{2.5} \log n)$ processing time.

Question

Can you get **distance-sparsifier** with $+1$ additive error?

Answer: No!



Related Upper/Lower bounds

Stretch	Size of Sparsifier	Remark
+2	$O(n^{1.5})$	[DHZ96]
+4	$O(n^{1.4})$	[BKMP05]
+6	$O(n^{1.33})$	[Che13]
+ c	$\Omega(n^{4/3})$	[AB16]

Table: Distance Sparsifiers

Exercises

Exercises

Compute

a sparsifier H with $+2$ additive error, and

$+2$ approximate distance matrix

in better than $n^{2.5}$ time.

Computing H in $O(n^{2.33})$ time

1. Max-deg in $P_{x,y}$ lies in $[1, \sqrt{n}]$
 - Add to H all edges incident to vertices of $\deg \leq \sqrt{n}$
2. Max-deg in $P_{x,y}$ lies in $[n^{2/3}, n - 1]$
 - Let R be random set of size $O(n^{1/3} \log n)$. For each $r \in R$, add to H edges of $BFS(r)$.
3. Max-deg in $P_{x,y}$ lies in $[\sqrt{n}, n^{2/3}]$
 - Let R^* be random set of size $O(\sqrt{n} \log n)$. For each $r \in R^*$, add to H edges of $BFS(r)$ computed in graph $G[\deg \leq n^{2/3}]$.

Exercise 1

Graph Sparsification

Compute a sparsifier H with $+2$ additive error in $O(n^2 \log^2 n)$ processing time.

Hint

For the range $[\sqrt{n}, n - 1]$, consider following $\log n$ partitions

$$[\sqrt{n}, 2\sqrt{n}], \dots, [2^i, 2^{i+1}], \dots, [\frac{n-1}{2}, n - 1].$$

Exercise 2

Distance approximation

Compute approximate distance matrix with $+2$ additive error in $O(n^{2.33})$ time.

Hint

Use the 3-level partitioning with additional ideas for BFS traversal.



Amir Abboud and Greg Bodwin.

The $4/3$ additive spanner exponent is tight.

In Daniel Wachs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 351–361. ACM, 2016.



Surender Baswana, Telikepalli Kavitha, Kurt Mehlhorn, and Seth Pettie.

New constructions of (α, β) -spanners and purely additive spanners.

In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 672–681. SIAM, 2005.



Shiri Chechik.

New additive spanners.

In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 498–512. SIAM, 2013.



Dorit Dor, Shay Halperin, and Uri Zwick.

All pairs almost shortest paths.

In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 452–461. IEEE Computer Society, 1996.